Portable Acceleration of CMS Production Workflow with Coprocessors as a Service

Mia Liu
On behalf of CMS collaboration
CPAD. Nov 7-10. 2023
FPGA-accelerated machine learning inference as a solution for particle physics computing challenges

Jennifer Ngadiuba, Maurizio Pierini (CERN)
Javier Duarte, Burt Holzman, Ben Kreis, Kevin Pedro, Mia Liu, Nhan Tran,
Aristeidis Tsaris (Fermilab)
Phil Harris, Dylan Rankin (MIT)
Zhenbin Wu (UIC)
Portable Acceleration of CMS Production Workflow with Coprocessors as a Service

The CMS Collaboration

Abstract

Computing demands for large scientific experiments, such as the CMS experiment at CERN, will increase dramatically in the next decades. To complement the future performance increases of software running on CPUs, explorations of coprocessor usage in data processing hold great potential and interest. We explore the approach of Services for Optimized Network Inference on Coprocessors (SONIC) and study the deployment of this as-a-service approach in large-scale data processing. In the studies, we take a data processing workflow of the CMS experiment as an example, and run the main workflow on CPUs, while offloading several machine learning (ML) inference tasks onto either remote or local coprocessors, such as GPUs. With experiments performed at Google Cloud, the Purdue Tier-2 computing center, and combinations of the two, we demonstrate the acceleration of these ML algorithms individually on coprocessors and the corresponding throughput improvement for the entire workflow. This approach can be easily generalized to different types of coprocessors, and deployed on local CPUs without throughput performance decrease. We emphasize that SONIC enables high coprocessor usage and enables the portability to run workflows on different types of coprocessors.
Heterogeneous Computing for ML/AI

Advances driven by big data explosion & machine learning

A 5 year old slide, message remains…

Discontinued: October 18, 2022
‘AI chips’ in 2023

- Meta
- Groq
- Graphcore
- Apple
- AMD / Xilinx
- Cerebras

GPUs have been the baseline for accelerated ML.

One of the big industry trends of recent years: custom silicon for AI.

- Mac M2: 15.8 TOPS Neural Engine
- Meta MTIA v1: 102.4 TOPS INT8, 51.2 TFLOPS FP16, 25 W
- Graphcore IPU mk2: 250 TFLOPS
- Groq: 750 TOPs INT8, 188 TFLOPS FP16
- Even your FPGA has accelerators - Versal AI: 145 TOPs INT8, 12 TFLOPS FP32

Also many examples from TinyML for ultra low power AI - e.g. GreenWaves GAP in MLPerf benchmark.
Services for Optimized Network Inference on Co-processors.


• Increasing demand for accelerating ML inference

• Demonstrated offloading ML inference to Microsoft Brainwave FPGA services
  • FPGA co-processor outperforms GPUs at batch-1 (relevant for streaming)

• CPU client software only handles preprocessing and I/O, not inference framework
Optimal Acceleration hardware usage

Traditional direct CPU->GPU connection:

- Too few models or cores = underutilized GPU
- Narrow “sweet spot” in terms of models or cores
- Too many models or cores = oversaturated GPU

Inflexible & Expensive

Complex, Requires R&D
Since 2018

Hardware platforms

GPU-as-a-service

FPGA-as-a-service Toolkit ('homegrown' gPRC server)

Open source tools

Deployment in experiments

This talk: Portable Acceleration of CMS Production Workflow with Coprocessors as a Service

GPU-as-a-service for DUNE

This talk: Portable Acceleration of CMS Production Workflow with Coprocessors as a Service

NVIDIA Triton Inference Server

Graphcore
How to deploy SONIC in CMS

NVIDIA Triton Inference Server
First demonstration in large experiment data processing

MiniAOD step of the CMS data processing: ML inferences consume 10% of the total processing time.

Tests at Purdue CMS Tier-2 data center, GCP and Fermilab.

Public results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time [ms]</th>
<th>Fraction [%]</th>
<th>Input [MB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN-AK4</td>
<td>42.4</td>
<td>4.3</td>
<td>0.04</td>
</tr>
<tr>
<td>PN-AK8</td>
<td>11.4</td>
<td>1.1</td>
<td>0.003</td>
</tr>
<tr>
<td>DeepMET</td>
<td>13.2</td>
<td>1.3</td>
<td>0.33</td>
</tr>
<tr>
<td>DeepTau</td>
<td>21.1</td>
<td>2.1</td>
<td>1.18</td>
</tr>
<tr>
<td>ParticleNet+DeepMET+DeepTau</td>
<td>88.1</td>
<td>8.8</td>
<td>1.55</td>
</tr>
<tr>
<td>Total</td>
<td>993.3</td>
<td>100.0</td>
<td>—</td>
</tr>
</tbody>
</table>

Mini-AOD production typically takes about 0.5 seconds per event on production grid nodes.
This includes learning the existing SONIC miniAOD workflow test setup at Purdue, with SONIC GPU servers managed by the Kubernetes services that are being developed by Dmitry. In the last quarter of 2023, Yao will start working on the proposed development in enhancing SONIC server-side support in customized non-ML GPU kernels of CMS reconstruction sequence. Detailed plans, milestones, and deliverables are presented below.

- **2023/Q4**: Automate the initial Patatrack-aaS workflow developed by developing wrap-pers for inputs consumed by the algorithms and outputs from the customized backend kernel, as well as an automatic wrapper from the client side. Benchmark the performance in latency, throughput, and memory consumption against inferences on directly connected GPUs. The deliverables are the automation tools developed and the performance test results.

- **2024/Q1**: Test the generalizability of the automation wrapper tools with other GPU reconstruction algorithms. These can include although not limited to the algorithms in the current CMS Run-3 GPU workflow and algorithms targeting the HL-LHC such as the Line segment tracking. The deliverables are to identify shortcomings in the automation tool in handling algorithms consuming different data formats and further improvements to be integrated into the SONIC automation tools.

- **2024/Q2**: This period is dedicated to developing and integrating the improvement in SONIC automation tools to successfully enable offloading of GPU algorithms in the CMS Run-3
Load-balanced Triton service available on site

- **CMS SONIC workflows**
  - Directed to directly connected GPU nodes
  - Sites with directly connected GPUs
    - GPU nodes excluded from SONIC server
    - Local host as SONIC server with GPU acceleration
  - Directed to sites without load balanced SONIC service
  - Sites with CPU only nodes
    - Local host as SONIC server without GPU acceleration

- **CMS SONIC workflows at Purdue Tier 2 data center**
  - Directed to Purdue Tier 2 data center where SONIC service available via Kubernetes
  - Directed to CMS SONIC GPU servers at Commercial clouds or HPC sites
  - Fall back to local CPU

- **SONIC GPU servers managed by Kubernetes**
  - Purdue Tier 2 data center
  - Results via gRPC
  - Launch on demand server
  - Results to CMS SONIC GPU servers

- **CMS SONIC GPU servers at Commercial clouds or HPC sites**
  - Access
  - NERSC
• Asynchronous inference requests

• Triton supports ONNX, TensorFlow, PyTorch, Scikit-Learn, etc. Triton model analyzer tool optimizes server settings. e.g. batch size, dynamic batching window.
**CPU/GPU ratio optimization**

One ML model offloaded for each test

Optimal CPU client jobs / GPU ratio decided by acceleration factor/saturation point

Scale-out test at GCP:

10,000 CPU cores (2,500 4-threaded) client jobs, 100 Tesla T4 GPUs with Kubernetes as load balancer

Peak network usage was ~15 GB/s (total bandwidth coming into GPU cluster)
Plan B?

CMS SONIC workflows

Directed to directly connected GPU nodes
- Sites with directly connected GPUs
- GPU nodes excluded from SONIC server
- Local host as SONIC server with GPU acceleration

Directed to sites without load balanced SONIC service
- Sites with CPU only nodes
- Local host as SONIC server without GPU acceleration

Directed to Purdue Tier 2 data center where SONIC service available via Kubernetes

CMS SONIC workflows at Purdue Tier 2 data center

SONIC GPU servers managed by Kubernetes
- Purdue Tier 2 data center

Results
- Launch on demand server

Fall back to local CPU

CMS SONIC GPU servers
- at Commercial clouds or HPC sites
Remote server/Fall-back CPU

No added latency observed in remote offloading.

More studies on memory overhead etc, see paper.
GPU as-a-service for analysis

NVIDIA Triton Inference-as-a-Service

- Share GPU resources for ML inference

• Integrated in python based columnar analysis framework, non experiment specific
Summary and Acknowledgement

Heterogeneous platforms as-a-service demonstrated with realistic workflow with NVIDIA Triton Service

Flexible & Adaptable - right-size the system based on compute needs, maximize e.g. GPU acceleration; task-based optimization;

Scalable & Portable - co-processor disassociated from existing CPU infrastructure; reduce client software dependency on server hardware

On-going: SONIC to offload customized algorithms in GPU (patatrk, algorithms written with Alpaka); Server-auto scaling; SONIC for AMD GPUs...

Promising computing paradigm for current and future online and offline high throughput systems

Thanks to support by NSF A3D3 HDR institute and USCMS operations program.